Segmentation and Labeling of Documents using Conditional Random Fields

Shravya Shetty, Harish Srinivasan and Sargur Srihari

TR-07-07 June 2007



Center of Excellence for Document Analysis and Recognition (CEDAR)
520 Lee Entrance, Suite 202
Amherst. New York 14228

This paper was also published in *Proc. Document Recognition and Retrieval IV, Proceedings of SPIE*, San Jose, CA, February 2007, pp. 6500U-1-11.

Segmentation and labeling of documents using Conditional Random Fields

Shravya Shetty[†], Harish Srinivasan[†], Matthew Beal and Sargur Srihari[†]

Center of Excellence for Document Analysis and Recognition (CEDAR)

{sshetty@cedar,hs32@cedar,srihari@cedar,mbeal@cse}.buffalo.edu

Department of computer science, University at Buffalo

State University of New York

ABSTRACT

The paper describes the use of Conditional Random Fields(CRF) utilizing contextual information in automatically labeling extracted segments of scanned documents as Machine-print, Handwriting and Noise. The result of such a labeling can serve as an indexing step for a context-based image retrieval system or a bio-metric signature verification system. A simple region growing algorithm is first used to segment the document into a number of patches. A label for each such segmented patch is inferred using a CRF model. The model is flexible enough to include signatures as a type of handwriting and isolate it from machine-print and noise. The robustness of the model is due to the inherent nature of modeling neighboring spatial dependencies in the labels as well as the observed data using CRF. Maximum pseudo-likelihood estimates for the parameters of the CRF model are learnt using conjugate gradient descent. Inference of labels is done by computing the probability of the labels under the model with Gibbs sampling. Experimental results show that this approach provides for 95.75% of the data being assigned correct labels. The CRF based model is shown to be superior to Neural Networks and Naive Bayes.

Keywords: Conditional Random Field(CRF); labeling scanned documents; handwritten text extraction

1. INTRODUCTION

Complex documents present a great challenge to the field of document recognition and retrieval. The combined presence of noise, handwriting, signature, logos, machine-print with different fonts, and rule lines impose a lot of restrictions to algorithms that work relatively well on simple documents. The primary task of processing these complex documents, is that of isolating the different contents present in the document. Once the contents such as handwriting, machine-print, signature and noise are separated out, they can now be called as indexed documents ready to be used by a context-based image retrieval system¹,² or for a Handwriting/Optical Character Recognition system,³ or for a signature verification system.⁴ This paper discusses a novel approach using Conditional Random Field(CRF) to label the different segments of a document as Machine-print, Handwriting/Signature or Noise.

Conditional Random Fields were originally proposed as a probabilistic framework for sequence labeling tasks and were found to best model the dependencies between the observed data⁵⁶.⁷ The motivation to use CRF for this application arises from the spatial inter-dependencies of the different regions in documents. For example, machine-print is sequential from left to right and so is handwriting. To label a segment without considering neighboring/contextual information sounds intuitively incorrect and has not led to satisfactory results. Previous work in trying to use contextual information was the use of Markov Random Fields⁸ as a final post processing step. The current paper discusses the use of Conditional Random Fields to model the entire problem at hand and it also helps relax the assumption of conditional independence of the observed segments of the document given their labels, as commonly used in Markov Random Fields. The problem is formulated as follows: Given a document, (i)Segment the document into a number of patches(approximately the size of a word), and (ii)Label each of the segments as one of Machine-print, Handwriting or Noise. The class of Handwriting includes those of Signatures and the class of Noise encompasses salt and pepper noise, scan noise, scratches, black borders and logos and the class of Machine-print includes printed text of different fonts. Figure 1 shows a sample document and the expected output at the end of the segmentation and the labeling process.

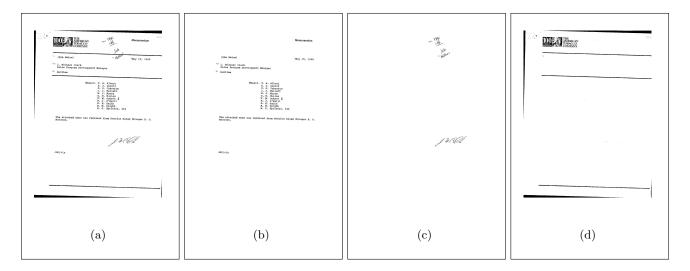


Figure 1. A sample business document from an archival database with the expected result of segmenting and labeling the document: (a) Original document, (b) Printed text, (c) Handwritten text and (d)Noise

The rest of the paper is organized as follows. Section 2 describes a generic CRF framework for the problem at hand. Section 3 describes a method for estimating the parameters for the model discussed in Section 2. Section 4 describes the implementation of a specific model for the problem at hand, including a description of the features, the method of inference of the labels and finally the performance results.

2. CONDITIONAL RANDOM FIELD MODEL DESCRIPTION

The probabilistic model of the Conditional Random Field used is given below.

$$P(\mathbf{y}|\mathbf{x},\theta) = \frac{e^{\psi(\mathbf{y},\mathbf{x};\theta)}}{\sum_{\mathbf{y}'} e^{\psi(\mathbf{y}',\mathbf{x};\theta)}}$$
(1)

where $\mathbf{y_i} \in \{\text{Machine-print}, \text{Handwriting}, \text{Noise}\}\$ and $\mathbf{x}: \text{Observed document and } \theta: \text{CRF model parameters. It}$ is assumed that a document is segmented into m non-overlapping patches (details of the segmentation algorithm are described later in section 4.1). Then

$$\psi(\mathbf{y}, x; \theta) = \sum_{j=1}^{m} \left(A(j, y_j, \mathbf{x}; \theta^s) + \sum_{(j,k) \in E} I(j, k, y_j, y_k, \mathbf{x}; \theta^t) \right)$$
(2)

The first term in equation 2 is called the state term(sometimes called Association potential as mentioned in⁶) and it associates the characteristics of that patch with its corresponding label. θ^s are called the state parameters for the CRF model. Analogous to it, the second term, captures the neighbor/contextual dependencies by associating pair wise interaction of the neighboring labels and the observed data(sometimes referred to as the interaction potential). θ^t are called the transition parameters of the CRF model. E is a set of edges that identify the neighbors of a patch. The specific set of neighbors used for this problem is described in detail in section 4.1. θ comprises of the state parameters, θ^s and the transition parameters, θ^t .

The association potential can be modeled as

$$A(j, y_j, \mathbf{x}; \theta^s) = \sum_i (h_i \cdot \theta_{ij}^{s_2})$$

where h_i is typically the state feature value associated with the patch being considered. In order to introduce a non-linear decision boundary we define h_i to be a transformed state feature vector

$$h_i = tanh\left(\sum_{l} (f_l^{s_1}(j, y_j, \mathbf{x}) \cdot \theta_l^{s_1} i)\right)$$

where f_l^s is the l^{th} state feature extracted for that patch and $\theta_{li}^{s_1}$ is the additional state parameter introduced here. The state features that are used for this problem are defined later in section 4.2 in table 1. The state features, f_l are transformed by the tanh function to give the feature vector \mathbf{h} . The transformed state feature vector can be thought analogous to the output at the hidden layer of a neural network. The state parameters θ^s are a union of the two sets of parameters θ^{s_1} and θ^{s_2} .

The interaction potential $I(\cdot)$ is generally an inner product between the transition parameters θ^t and the transition features f_t . To introduce non-linearity, we use the idea of kernels, and the interaction potential is defined as follows:

$$I(j, k, y_j, y_k, \mathbf{x}; \theta^t) = \sum_{l} (\phi_l \cdot \theta_l^t)$$

where ϕ_l is the l^{th} transition feature after applying a quadratic kernel on the original transition features as defined below.

$$\Phi_l = \langle f^t(j, k, y_j, y_k, \mathbf{x}) \cdot f^t(j, k, y_j, y_k, \mathbf{x}) \rangle$$

3. PARAMETER ESTIMATION

There are numerous ways to estimate the parameters of this CRF model. In order to avoid the computation of the partition function we learn the parameters by maximizing the pseudo-likelihood of the documents, which is an approximation of the maximum likelihood value. For this paper, we estimate the Maximum pseudo-likelihood parameters using conjugate gradient descent with line search. The pseudo-likelihood estimate of the parameters, θ are given by equation 3

$$\hat{\theta_{ML}} \approx \arg\max_{\theta} \prod_{i=1}^{M} P(y_i | y_{\mathcal{N}_i}, \mathbf{x}, \theta)$$
 (3)

where $P(y_i|y_{\mathcal{N}_i}, \mathbf{x}, \theta)$ (Probability of the label y_i for a particular patch i given the labels of its neighbors, $y_{\mathcal{N}_i}$), is given below.

$$P(y_i|y_{\mathcal{N}_i}, \mathbf{x}, \theta) = \frac{e^{\psi(y_i, \mathbf{x}; \theta)}}{\sum_a e^{\psi(y_i = a, \mathbf{x}; \theta)}}$$
(4)

where $\psi(y_i, x; \theta)$ is defined in equation 2.

Note that the equation 3 has an additional $y_{\mathcal{N}_i}$ in the conditioning set. This makes the factorization into products feasible as the set of neighbors for the patch form the minimal Markov blanket. It is also important to note that the resulting product only gives a pseudo-likelihood and not the true likelihood. The estimation of parameters which maximize the true likelihood may be very expensive and intractable for the problem at hand.

From equation 3 and 4, the log pseudo-likelihood of the data is

$$\mathcal{L}(\theta) = \sum_{i=1}^{M} \left(\psi(y_i = a, x; \theta) - \log \sum_{a} e^{\psi(y_i = a, x; \theta)} \right)$$

Taking derivatives with respect to θ we get

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = \sum_{i=1}^{M} \left(\frac{\partial \psi(y_i, x; \theta)}{\partial \theta} - \sum_{a} P(y_i = a | y_{\mathcal{N}_i}, x, \theta) \cdot \frac{\partial \psi(y_i = a, x; \theta)}{\partial \theta} \right)$$

The derivatives with respect to the state and transition parameters are described below. The derivative with respect to parameters θ^{s_2} corresponding to the transformed features $h_i(j, y_j, \mathbf{x})$ is given by

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_{iu}^{s_2}} = \sum_{i=1}^{M} h_i(y_j = u) - \sum_{a} P(y_j = a | y_{\mathcal{N}_i}, x, \theta) h_i(a = u)$$

Here, $h_i(y_j = u) = h_i$ if the label of patch j is u otherwise $h_i(y_j = u) = 0$ The derivative with respect to parameters θ^{s_1} corresponding to the state features $f_l(j, h_i, \mathbf{x})$ is given by

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_{li}^{s_1}} = \sum_{i=1}^{M} (\theta_{iy_j}^{s_2} (1+h_i)(1-h_i) f_l - \sum_{a} P(y_j = a|y_{\mathcal{N}_i}, x, \theta) \theta_{iy_j}^{s_2} (1+h_i)(1-h_i) f_l)$$

Similarly, the derivative with respect to the transition parameters, θ_t is given by

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta_{l_{cd}}^t} = \sum_{j=1}^M \left(\sum_{j,k \in E} \phi_l(y_j = c, y_k = d) - \sum_a \sum_{j,k \in E_{cd}} P(y_j = a | y_{\mathcal{N}_i}, x, \theta) \phi_l(a = c, y_k = d) \right)$$

4. EXPERIMENTS AND RESULTS

The Tobacco industrial litigation archives were used for the evaluation of the model described above. Figure 1(a) shows a sample scanned document from the dataset. A total of 53 such documents were used for these experiments. Each document provides for a number of patches, typically an average of 150 patches exist for a full sized document. The algorithm for obtaining these patches is described in section 4.1. The state and transition features, defined later in 4.2, are then computed for each of these patches. The 53 documents provide for a total of 7500 such patches. 3700 patches from 26 of these documents were used as the training set to estimate the parameters for the CRF model as described in section 3. The remaining 3800 patches from 27 documents were used as the test set. The overall accuracy of the correctly labeled data was 95.75%. More detailed results are mentioned in Section4.4. The operational steps in labeling scanned documents are shown in Figure 2.

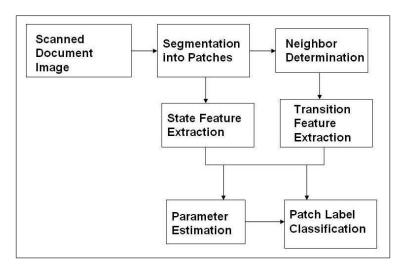
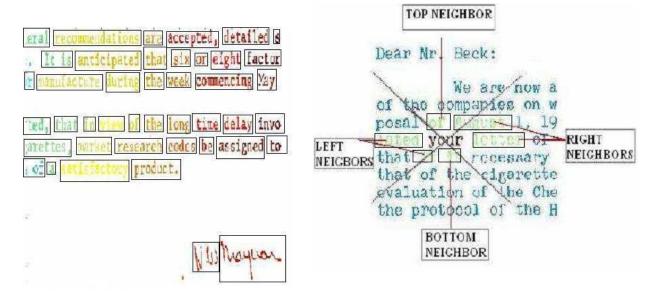


Figure 2. A bock diagram describing the steps involved in labeling scanned documents

4.1. Segmentation and Neighbor determination

A patch is defined to be a region in a document such that, if a rectangular window(size determined dynamically for each document) is drawn with each foreground pixel within the patch at its center, then the window shall not contain any foreground pixel from another patch. The algorithm for generating these patches is a region growing algorithm and a brief description is given below.

- 1. Initialize every pixel to be a separate patch.
- 2. Start with a foreground pixel that is not already marked.



- (a) A region of a sample document showing the result of the segmentation process. Each patch has been marked with a box enclosing it.
- (b) A region of a sample document showing a patch with the word "'your"' and its six neighbors

Figure 3. (a) Segmentation on sample document (b) Neighbors for a segment(patch)

- 3. With this pixel as the center, draw a rectangular window around it. The size of the window is optimized to get the desired size of patches on a validation set of documents.
- 4. All foreground pixels of connected components with any pixel enclosed within this rectangular window are marked as belonging to the same patch as that of the center pixel.
- 5. Repeat steps 2 through 4 until all pixels are marked.
- 6. Patches with fewer pixels than a predetermined threshold are ignored as noise and are not attempted to be labeled as one of machine-print, handwriting/signature, noise. It is important to note here that patches having more pixels than this threshold pixels could still be noise.

Figure 3(a) shows the result of segmentation on a small region of a document. Each patch is marked with a box around it. The size of the patch was optimized in a way to represent approximately the size of a word. Once all the patches are obtained for a document, the neighboring patches are identified. A total of 6 neighbors are identified for each patch. The 6 neighbors include 1 on the top, 1 on the bottom, 2 to the right, and 2 to the left of the patch. These neighbors are the closest(top/bottom) and the two closest(left/right) in terms of the convex-hull distance between the patches considered. More neighbors are included from the right and left of the patch since scanned documents have greater dependency across the width of the document. For example, machine printed and handwritten text are written from left to right. The definitions of top, bottom, left and right are determined from the center of gravity of the patch being considered. However the convex-hull distance between two patches is measured taking the entirety of both the patches. Figure 3(b) shows the patch and its neighbors. It is planned to experiment with more flexible definitions for neighbors in the future.

4.2. Feature extraction

State features as defined previously in section 2 try to associate each patch to a label using the characteristics of that patch alone. Analogous to these, transition features associate a patch to a label using information from

the neighboring patches. 23 state features, are extracted for each patch and these are described in Table 1. 4 Transition features, described in Table 2, are then computed for every pair of neighboring patches using the state features and the neighbor label information. Using these extracted features from each of the 3500 patches in the training set, the parameters of the CRF are estimated using conjugate gradient descent, as described in Section 3.

State Feature	Description		
Height	Maximum height of the patch		
Avg component width	The mean width of the connected components within a patch		
Density	Density of foreground pixels within the patch		
Aspect ratio	Width/Height of the patch		
Gabor filter	8 features capturing the different stroke orientations		
Variation of height	Variation in height within a patch		
Width variation	Variation in width within a patch		
Overlap	The total overlap in area between the connected components within a patch		
Percentage of text above	Relative location of the patch with respect to the entire document		
Number of components	Count of the connected components within a patch		
Maximum component size	Maximum size of a component within a patch		
Points in convex hull	Number of points in the convex hull of a patch		
Maximum run length	The maximum horizontal run length within a patch		
Avg run length	The average horizontal run length within a patch		
Horizontal Transitions	A count of the number of times the pixel value transitions from white to		
	black horizontally		
Vertical Transitions	A count of the number of times the pixel value transitions from white to		
	black vertically		

Table 1. Description of the 23 state features used

Transition Feature	Description
Relative location	Assigned weights based on the relative location - top/bottom or right/left
Convex hull distance	The convex hull distance between the 2 patches
Ratio of aspect ratio	The ratio of the aspect ratio values of the 2 patches
Ratio of number of components	The ratio of the number of components present in the 2 patches

Table 2. Description of the 4 transition features used - The transition features are computed for a patch and its neighbor.

4.3. Inference

The goal of inference is to assign a label to each of the patches being considered. The algorithm for inference uses the idea of Gibb's sampling.¹⁰

- 1. Randomly assign labels to each of the patches in a document based on an intuitive prior distribution of the labels.
- 2. Choose a patch at random and compute the probability of assigning each of the labels using the model from the equation given below to obtain a probability distribution p for the labels.

$$P(y_i|y_{\mathcal{N}_i}, \mathbf{x}, \theta) = \frac{e^{\psi(y_i, \mathbf{x}; \theta)}}{\sum_a e^{\psi(y_i = a, \mathbf{x}; \theta)}}$$
(5)

- 3. Use Gibbs sampling to sample from this distribution p to assign a probable label to the patch.
- 4. Repeat steps 2 and 3 until the assignments do not change. Store the set of label assignments along with the probability distribution p.

- 5. Repeat steps 1-4, for a sufficient number of iterations in order to eliminate the dependency on the initial random label assignments.
- 6. Consider the set of arrived assignments at step 4 in each of the iterations, and for all the patches pick the labels with the maximum probability as the final set of labels.

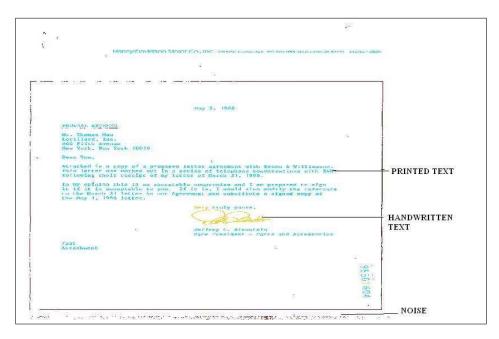


Figure 4. A sample document with its patches labeled as machine printed/handwritten/noise

4.4. Results

The results presented here are on 27 randomly chosen documents comprising of about 4000 patches. Table 3 shows the results of the labeling accuracy(Recall) and the Precision values on these patches. Recall and Precision are defined in terms of the amount of text which refers to a count of the foreground pixels(made resolution independent).

Recall of label '
$$\mathbf{a}' = \frac{Amount\ of\ correctly\ classified\ data\ of\ label\ '\mathbf{a}'}{Total\ amount\ of\ data\ of\ label\ '\mathbf{a}'}$$

$$Precision\ of\ label\ '\mathbf{a}' = \frac{Amount\ of\ correctly\ classified\ text\ of\ label\ '\mathbf{a}'}{Total\ amount\ of\ text\ classified\ to\ be\ of\ label\ '\mathbf{a}'}$$

An overall accuracy of 95.75% and individual class accuracies of 98%, 95% and 90% was obtained. Figure 4 shows the result of labeling a sample document.

The performance of our CRF based approach was compared to other methods like Neural Networks and Naive Bayes. In case of the neural network and naive bayes approach the State features described in section 4.2 were used for classification. The neural network had 3 output neurons and 8 hidden neurons. The parameter learning was done using back propagation with an objective of minimizing the overall error rate. The Naive Bayes classifier learns a normal distribution independently for each State feature for each of the label types. A comparison of the accuracies of these 3 methods is shown in Table 4. This comparison shows that our CRF approach is significantly better than the other two methods. In case of the neural network classifier the accuracy of the machine printed text is almost the same as our classifier but the performance for the other two label types is significantly worse. Due to large amount of printed text present in the test documents the overall accuracy

Type of Label	Precision	Recall
Machine Printed Text	96.29%	98.36%
Handwritten Text	90.18%	94.81%
Noise	95.72%	89.80%
Overall	-	95.75%

Table 3. The results of labeling the documents

	CRF	Neural Network	Naive Bayes
Machine Printed Text	1.64%	2.35%	11.54%
Handwritten Text	5.19%	20.90%	25.04%
Noise	10.20%	15.00%	12.23%
Total	4.25%	7.04%	12.58%

Table 4. Comparison of error rates using three approaches

is 7.19% inspite of the poor performance of the other two labels. In case of the Naive Bayes classifier, both the overall accuracy and the accuracy of each of the three labels is low due to the assumption of indepedence of the features made. The CRF based model results in a higher performance as it utilizes the spatial inter-dependencies of neighboring patches in labeling them.

An approximate comparison of our model with the Markov Random Field approach⁸ can be done by considering the labeling accuracy at the patch level. 99.1% of the patches were correctly labeled as opposed to the 98% reported in.⁸ Note that the algorithm for segmentation is different in the two implementations, enabling only an approximate comparison.

5. CONCLUSION

A novel use of Conditional Random Fields to segment and label documents was proposed and discussed. The model exploits spatial inter-dependencies inherent in the data by using features from neighboring patches in addition to features specific to the current patch. 95.75% of the data were assigned the correct labels with individual label accuracies of 98%, 95% and 90% corresponding to machine print, handwriting and noise respectively. The resulting labelled documents can be effectively used in content based image retrieval, Optical Handwriting Recognition and signature based biometric verification. The performance of the model was shown to be superior to methods using Neural Networks and Naive Bayes.

REFERENCES

- 1. S. Srihari, S. Shetty, S. Chen, H. Srinivasan, and C. Huang, "Document image retrieval using signatures as queries," *Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06)*, 2006.
- 2. S. Ravela and R. Manmatha, "Image retrieval by appearance," ACM SIGIR Forum, pp. 278–285, 1997.
- 3. S. N. Srihari, A. Ganesh, C. Tomai, Y.-C. Shin, and C. Huang, "Information retrieval system for handwritten documents," *Document Analysis Systems VI Proc. 6th International Workshop, DAS*, pp. 298–309, 2004.
- 4. H. Srinivasan, S. N. Srihari, and M. Beal, "Signature verification using kolmogorov-smirnov statistic," *Proc. International Graphonomics Society Conference (IGS)*, pp. 152–156, 2005.
- J. Lafferty, A. Macullum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequential data," Eighteenth international conference on Machine Learning(ICML-2001), pp. 282–289, 2001.
- 6. S. Kumar and M. Hebert, "Discriminative fields for modeling spatial dependencies in natural images," Advances in Neural information processing systems (NIPS-2003), 2003.
- 7. A. Quattoni, M. Collins, and T. Darrel, "Conditional random fields for object recognition," Advances in Neural Information Processing Systems 17 (NIPS 2004), 2005.
- 8. Y. Zheng, H. Li, and D. Doermann, "Machine printed text and handwriting identification in noisy document images," *IEEE Trans. Pattern Analysis Machine Intelligence* vol.26(3), 2003.

- 9. H. Wallach, "Efficient training of conditional random fields," *Proc. 6th Annual CLUK Research Colloquium* , 2002.
- 10. G. Casella and E. George, "Explaining the gibbs sampler," Amer. Statistician vol.46, pp. 167–174, '1992.